Points fixes et sémantique

Le langage IML_{rep}

Le langage IML_{rep} est une variante du langage IML. Les classes syntaxiques $\mathcal{A}exp$ et $\mathcal{B}exp$ sont les mêmes du langage IML. La classe $\mathcal{C}om$ des commandes est définie par la grammaire

$$c = \ X{:=}a \mid c_0\ ;\ c_1 \mid \texttt{if}\ b\ \texttt{then}\ c\ \texttt{else}\ c \mid \texttt{repeat}\ c\ \texttt{until}\ b$$

où $a \in \mathcal{A}exp$, $b \in Be$.

Exercice 1. Définissez la sémantique opérationnelle, et dénotationnelle de ce langage. (Il suffira définir la sémantique de la boucle `repeat`).

Exercice 2. Dans le langage IML_{rep} peut-on définir une commande qui se comporte comme la commande `skip` ? Si oui, en donner une preuve formelle à l'aide de la sémantique.

Le plus petit point préfixe

Exercice 3. Soient $f : L \longrightarrow M$ et $g : M \longrightarrow L$ deux fonctions monotones, où L et M sont deux treillis complets. On connaît, donc, qu'un plus petit point fixe de $g \circ f$, noté $fix(g \circ f)$, et un plus petit point fixe de $f \circ g$, noté $fix(f \circ g)$, existent. Montrer que :

$$f(\ fix(g \circ f)\ ) = fix(f \circ g)\ .$$

On peut démontrer à l'aide des axiomes suivants :

$$\begin{aligned} h(fix(h)) &\leq fix(h) \\ h(y) \leq y \ \ \Rightarrow\ \ fix(h) &\leq y\ . \end{aligned}$$

Exercice 4. On considère maintenant un langage IML^{+}_{rep} : les classes syntaxiques $\mathcal{A}exp$ et $\mathcal{B}exp$ sont toujours les mêmes, la classe $\mathcal{C}om$ des commandes inclue des boucles `while` ainsi que des boucles `repeat`. Sa grammaire est :

$$c := X := a \mid c_0\ ;\ c_1 \mid \texttt{if}\ b\ \texttt{then}\ c_0\ \texttt{else}\ c_1 \mid \texttt{repeat}\ c\ \texttt{until}\ b \mid \texttt{while}\ b\ \texttt{do}\ c$$

où $a \in \mathcal{A}exp$, $b \in \mathcal{B}exp$, et $c, c_0, c_1 \in \mathcal{C}om$.
On suppose que vous avez proposé la « bonne » sémantique dénotationnelle à la boucle `repeat`. Démontrer alors l'équivalence des commandes

$$\texttt{repeat}\ c\ \texttt{until}\ b \qquad \texttt{c;}\ \texttt{while}\ \texttt{not}\ b\ \texttt{do}\ c\ .$$

Suggestions :

1. Si $P \subseteq \mathcal{S}$, on définit la relation binaire

   $$T_P = \{\ (\sigma, \sigma) \,|\, \sigma \in P\ \}\ .$$

   Montrer alors que

   $$R \circ T_P = \{\ (\sigma, \sigma') \,|\, \sigma \in P \text{ et } (\sigma, \sigma') \in R\ \}\ .$$

   Déterminer aussi $T_P \circ R$.
2. Pour $b \in \mathcal{B}exp$, soit $T_b = \{\,(\sigma, \sigma) \mid (\sigma, 1) \in \|b\|_{\mathcal{B}exp}\,\}$. Exprimer l'opérateur $\Gamma(Z)$ – dont le point fixe interprète la commande `while` b `do` c – à l'aide des relations T_b, $\gamma = \|c\|_{\mathcal{C}om}$ et des opérations d'union et de composition.
   Faire de même pour l'opérateur $\Delta(Z)$ dont le point fixe interprète la commande `repeat` c `until` b.
3. Se convaincre que la composition des relations est distributive par rapport à la union :

   $$\begin{aligned} R \circ (S_1 \cup S_2) &= (R \circ S_1) \cup (R \circ S_2) \\ (R_1 \cup R_2) \circ S &= (R_1 \circ S) \cup (R_2 \circ S) \end{aligned}$$

4. On pourra maintenant se servir de l'Exercice 3.